

## ALGORITMUL FURNICII

Literatura de specialitate menționează numeroase și variate definiții ale noțiunii de agent inteligent. De exemplu, unii autori consideră că sistemele de filtrare a poștei electronice sunt agenți inteligenți. Cu toate acestea, majoritatea cercetătorilor sunt de acord că elementul central al unui agent inteligent este adaptabilitatea sau capacitatea de a învăța. Ca urmare, o definiție concisă ar putea fi următoarea: un agent inteligent este o entitate care percepe, învață și acționează într-un anumit mediu.

Câteva dintre proprietățile remarcabile ale agenților inteligenți sunt următoarele:

- **autonomia** descrie capacitatea agenților de a lua decizii sau de a face alegeri pe baza propriei experiențe, folosind și informații și cunoștințe din și despre mediul în care acționează;
- **adaptabilitatea** descrie capacitatea agenților ca pe măsură ce reacționează la stimuli sau interacționează cu mediul exterior, să învețe, îmbunătățindu-și performanțele în timp;
- **capacitatea de colaborare** asigură partajarea cunoștințelor și experiențelor consumate cu alți agenți;
- **mobilitatea** descrie capacitatea agenților de a se deplasa în mediul exterior pe baza deciziei proprii.

Se poate spune că un agent care răspunde cu succes tuturor acestor cerințe este un agent inteligent dacă realizează întotdeauna acea acțiune care maximizează indicatorul său de performanță, în contextul unei secvențe de cunoștințe avută la dispoziție până în acel moment.

Una din direcțiile de dezvoltare ale agenților inteligenți, care a fost inițiată la începutul anilor 1990, este cea cunoscută sub numele de *modelul de optimizare după mușuroiul de furnici* sau, pe scurt, *algoritmul furnicii* (AF). Cercetătorii din domeniul etiologiei și științelor comportamentale la animale au elaborat o serie de modele care să explice unele aspecte particulare ale comportării sociale a insectelor, cum ar fi de exemplu procesul de auto-organizare. Ulterior, inspirându-se din aceste modele, alți cercetători au elaborat o serie de algoritme specializate pentru rezolvarea unor probleme dificile din punctul de vedere al efortului de calcul.

Prima formă a AF a fost propusă de Marco Dorigo în 1992, autorul considerând că acest algoritm face parte dintr-un concept mai larg, cel al *inteligenței multîmilor*. (în engleză, *swarm intelligence*).

### Principiul de funcționare al algoritmului furnicii

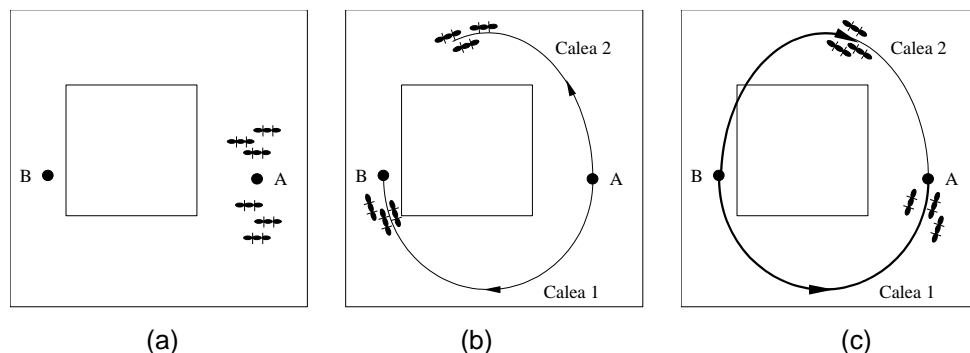
Dacă se urmărește comportarea furnicilor în natură, se constată că acestea pot găsi drumul cel mai scurt de la adăpost la sursa de hrană în absența oricăror informații vizuale și fără o comunicare directă între ele. De asemenea, aceleași furnici sunt apte să se adapteze la schimbările mediului înconjurător. AF încearcă să folosească aceste abilități ale furnicilor reale în scopul rezolvării anumitor tipuri de probleme de optimizare.

Urmărind o colonie de furnici, se constată că fiecare din ele pare să se miște la întâmplare, fără un scop precis. Totuși, deodată, grupuri din ce în ce mai mari de furnici se concentrează pe direcții comune. Această comportare pare să indice formarea unei inteligențe colective, ce are la bază comportarea fiecărui individ.

În natură, există o multitudine de căi pe care furnicile ajung să formeze o inteligență colectivă. Dintre acestea, AF folosește capacitatea furnicilor de a elibera *feromoni*, lăsând o urmă *invizibilă* a traseului lor. Feromonii sunt substanțe chimice produse de un animal și servesc, în principal, ca stimuli pentru alți indivizi din aceeași specie, pentru anumite răspunsuri comportamentale.

Pentru a înțelege cum folosesc furnicile feromonii pentru căutarea drumului de lungime minimă între două puncte oarecare, în prezența obstacolelor, se consideră cazul a două grupuri de furnici, având același număr de membri (de exemplu, câte 3 furnici de fiecare grup). Cele două grupuri trebuie să se deplaseze între punctele A și B, unde A reprezintă adăpostul sau locul de depozitare a hranei, iar B – sursa de hrană (Fig. 1.a). Trebuie spus de la bun început că, plecând din A, furnicile nu-și cunosc destinația, deplasarea lor având loc, fără o țintă prealabilă, în căutarea hranei. Admitem însă că singurul loc din vecinătatea adăpostului A în care se găsește hrană este punctul B.

În mod natural, furnicile vor căuta traseul pentru care deplasarea de la A la B și înapoi la A să se facă în timpul cel mai scurt. Desigur, acest timp va corespunde distanței minime dintre punctele A și B. Furnicile nu cunosc lungimile celor două căi între punctele A și B, însă cele două grupuri de furnici pornesc simultan din A, pe *Calea 1* și *Calea 2*, către B. Deoarece *Calea 1* este mai scurtă, primul grup va ajunge în B înaintea celui de-al doilea (Fig. 1.b). Pe acest drum cele trei furnici lasă o urmă de feromoni într-o concentrație proporțională cu numărul lor, la fel cum se întâmplă și cu furnicile din cel



**Fig. 1** Cum folosesc furnicile feromonii pentru stabilirea traseului de lungime minimă.

de-al doilea grup. După ce au luat hrana din punctul B, furnicile primului grup pornesc pe drumul de întoarcere spre A, folosind aceeași cale, și ajung în A la un moment în care cel de-al doilea grup de furnici se află undeva pe *Calea 2*, la dus sau la întors (nu contează). Pe drumul de întoarcere furnicile din primul grup lasă o nouă urmă de feromoni, care se suprapune peste prima, astfel încât concentrația de feromoni pe *Calea 1* se dublează (linia groasă din Fig. 1.c). În acest moment, din A pleacă alte grupuri de furnici care, dintre căile posibile (în general, pot fi oricâte, nu doar două ca în exemplul din Fig. 1), vor alege-o pe cea pe care concentrația de feromoni este maximă, adică *Calea 1* (altfel spus, calea de lungime minimă). Astfel, fără a utiliza inteligență individuală, furnicile reușesc să găsească calea de lungime minimă și să formeze o inteligență colectivă.

Caracterul dus-întors al drumului urmat de furnici este esențial în formarea acestui comportament, deoarece în momentul plecării din adăpost furnicile nu cunosc distanța pe care o au de parcurs, iar în momentul în care ajung la sursa de hrană nu pot transmite localizarea acesteia celorlalte furnici. Primele furnici plecate în căutarea hranei joacă de fapt rolul unor *cercetași*. În ipoteza unui teren „virgin”, fără urme de feromoni, acești cercetași își aleg căile la întâmplare. Atât timp cât nici unul din cercetași nu a parcurs drumul complet, înapoi la adăpost, alte furnici care ar porni pe urmele cercetașilor ar alege la întâmplare una din căile deschise de aceștia, în mod echiprobabil, deoarece concentrația de feromoni este aceeași pe toate aceste căi. Abia în momentul în care unul dintre cercetași revine la adăpost, concentrația de feromoni de pe calea urmată de acesta se dublează, iar dintre furnicile ce pornesc la drum, o proporție mai mare va alege această cale, contribuind la creșterea suplimentară a concentrației de feromoni. În acest fel, treptat, un număr tot mai mare de furnici vor parcurge drumul dintre adăpost și sursa de hrană și înapoi pe calea de lungime minimă deschisă de primul cercetaș.

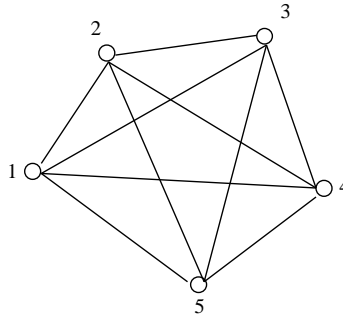
Copierea acestui comportament natural în cadrul unui algoritm artificial se face, de regulă, diferit, în funcție de tipul de problemă de optimizare ce se dorește a fi rezolvată. De asemenea, implementarea AF introduce o serie de modificări în raport cu modelul natural, modificări care au, în general, un caracter simplificator. De exemplu, deoarece distanțele dintre diferite puncte din spațiul problemei sunt mărimi cunoscute, nu mai este necesară ipoteza drumului dus-întors, iar echivalentul concentrației de feromoni din modelul natural, pentru calea respectivă, se alege invers proporțional cu lungimea acelei căi.

### Implementarea algoritmului furnicii pentru problema comisului voiajor

Problema comisului voiajor este una dintre cele mai cunoscute probleme de optimizare și se enunță astfel: un comis voiajor trebuie să viziteze o serie de clienți ce locuiesc într-un anumit număr de orașe. Se dorește să se stabilească ordinea în care trebuie vizitate orașele astfel încât lungimea totală a itinerarului parcurs de comis voiajor să fie minimă.

Se consideră un graf complet ce conține  $N$  noduri (orașele vizitate de comisul voiajor) și toate laturile ce pot fi definite între aceste noduri (Fig. 2). Implementarea AF pentru problema considerată este prezentată în Caseta 1.

Fiecărei laturi  $(i,j)$  din graful complet  $i$  se asociază o concentrație de feromoni  $\tau_{ij}$ , care va fi folosită pentru selectarea traseului urmat de colonia de furnici, care în continuare vor fi denumite *agenți*. Inițial, concentrațiile  $\tau_{ij}$  se stabilesc la valori pozitive mici (de exemplu,  $c = 0.01$ ).



**Fig. 2** Graful complet pentru problema comisului voiajor.

**Caseta 1** Algoritmul furnicii pentru problema comisului voiajor.

1. Date inițiale:  $N$  – numărul de orașe;  $NA$  – numărul de agenți;  $(x_i, y_i)$ ,  $i = 1, \dots, N$  – coordonatele orașelor;  $c$  – concentrația de feromoni pentru inițializarea probabilităților;  $Q$  – cantitatea totală de feromoni ai unui agent;  $\rho$  – factorul de evaporare;  $T_{max}$  – numărul maxim de iterații.
2. Inițializarea concentrațiilor de feromoni pe laturi și calculul distanțelor dintre noduri.
 

```

      for  $i = 1$  to  $N$  do
        for  $j = 1$  to  $N$  do
          if  $i \neq j$  then  $\tau_{ij} = c$ 
          if  $i < j$  then  $d_{ij} = \text{sqrt}((x_i - x_j)^2 + (y_i - y_j)^2)$ 
          else if  $i > j$  then  $d_{ji} = d_{ij}$ 
      
```
3. Amplasarea agenților în noduri (câte  $NA / N$  agenți în fiecare nod).
 

```

       $m = NA / N$ 
      for  $k = 1$  to  $NA$  do
         $o = (k - 1) \text{ div } m$  // Partea întreagă a câtului
         $Oraș_k = o + 1$ 
      
```
4. Inițializări: traseul de lungime minimă ( $LMin = \infty$ ) și contorul de iterații ( $t = 1$ ).
5. Inițializarea poziției în listele tabu ( $s = 1$ ).
6. Inserarea primului oraș în lista tabu a fiecărui agent.
 

```

      for  $k = 1$  to  $NA$  do  $Tabu_k(s) = Oraș_k$ 
      
```
7. Formarea traseelor (completarea listelor tabu).
 

```

      repeat
         $s = s + 1$  // Incrementarea poziției în listele tabu
        for  $k = 1$  to  $NA$  do // Pentru fiecare agent ...
           $i = Oraș_k$  // Orașul de origine
          // Probabilitățile de deplasare în orașele vecine
           $Sum = 0$ 
          for  $j = 1$  to  $N$  do
            if  $j \notin Tabu_k$  then  $Sum = Sum + (\tau_{ij})^\alpha \cdot (1 / d_{ij})^\beta$ 
          for  $j = 1$  to  $N$  do
            if  $j \notin Tabu_k$  then  $P_{ij} = (\tau_{ij})^\alpha \cdot (1 / d_{ij})^\beta / Sum$ 
            else  $P_{ij} = 0$ 
          // Selectarea orașului de deplasare
           $P_{max} = 0$ ;  $j^* = 0$ 
          for  $j = 1$  to  $N$  do
            if  $j \notin Tabu_k$  then
              if  $P_{ij} > P_{max}$  then
                 $P_{max} = P_{ij}$ 
                 $j^* = j$ 
           $Oraș_k = j^*$  // Mutarea agentului  $k$  în nodul  $j^*$ 
           $Tabu_k(s) = j^*$  // Înscrierea nodului  $j^*$  în lista tabu
      until  $s = N$ 
      
```
8. Închiderea traseelor agenților (mutarea fiecărui agent în orașul de origine).
 

```

      for  $k = 1$  to  $NA$  do  $Tabu_k(s+1) = Tabu_k(1)$ 
      
```

```

9.  Calculul lungimii totale a traseului fiecărui agent.
    for  $k = 1$  to  $NA$  do
       $L_k = 0$ 
      for  $p = 1$  to  $s$  do
         $i = Tabu_k(p)$ 
         $j = Tabu_k(p+1)$ 
         $L_k = L_k + d_{ij}$ 
10. Memorarea traseului de lungime minimă.
    for  $k = 1$  to  $NA$  do
      if  $L_k < L_{min}$  then  $L_{min} = L_k$ 
11. Actualizarea concentrațiilor de feromoni pe laturile grafului.
    for  $k = 1$  to  $NA$  do
      // Calculul corecției concentrației de feromoni
      for  $p = 1$  to  $s$  do
         $i = Tabu_k(p)$ 
         $j = Tabu_k(p+1)$ 
         $\Delta\tau_{ij} = \Delta\tau_{ij} + Q / L_k$ 
        // Modificarea concentrației de feromoni
      for  $i = 1$  to  $N$  do
        for  $j = 1$  to  $N$  do
          if  $i \neq j$  then  $\tau_{ij} = \rho \cdot \tau_{ij} + \Delta\tau_{ij}$ 
12. Criteriul de oprire: dacă  $t = T_{max}$ , algoritmul se încheie; în caz contrar, se golesc
    listele tabu:
    for  $k = 1$  to  $NA$  do
      for  $p = 1$  to  $s+1$  do
         $Tabu_k(p) = 0$ 
    și se revine la pasul 5.

```

Deoarece problema urmărește determinarea unei valori minime, lungimea traseului minim se inițializează cu o valoare foarte mare, notată simbolic  $LMin = \infty$ . În continuare, cei  $NA$  agenți se distribuie uniform (dacă este posibil) între nodurile din graf. Dacă se admite că numărul de noduri și numărul de agenți se aleg astfel încât  $NA = m \cdot N$  (unde  $m$  este un întreg oarecare), în fiecare nod se vor distribui  $m = NA / N$  agenți. Nodul în care se află la un moment dat fiecare agent  $k$  este notat  $Oras_k$ .

Având în vedere că problema conține restricția conform căreia fiecare agent trebuie să treacă prin fiecare oraș fără a forma cicluri (un oraș se vizitează o singură dată, cu excepția primului oraș, în care se încheie turul), memorarea traseului fiecărui agent se face într-o listă tabu, care va conține  $N + 1$  elemente, care descriu succesiunea în care sunt vizitate orașele. De exemplu, dacă cele 5 orașe din Fig. 1 sunt vizitate de un agent în ordinea 3, 1, 4, 5 și 2, lista tabu a acestuia va avea forma (3, 1, 4, 5, 2, 3).

După distribuirea agenților în nodurile grafului, lista tabu asociată fiecărui agent este inițializată în prima poziție cu numărul de ordine al orașului în care a fost distribuit acel agent. În continuare, urmează ca agenții să se deplaseze în diferite noduri ale grafului, până la completarea tuturor listelor tabu, moment în care fiecare agent va fi încheiat un tur complet al grafului. Astfel, pentru fiecare agent  $k$ , se consideră orașul de origine  $i = Oras_k$  și se alege orașul destinație  $j^*$ , care nu trebuie să figureze în lista tabu a agentului  $k$ , pe baza unor probabilități calculate în funcție de concentrația de feromoni de pe laturile  $(i, j)$  și vizibilitatea nodurilor  $j$  „privite” din nodul  $i$ , apreciată prin inversul distanței dintre noduri,  $d_{ij}$ . Aceste probabilități se calculează cu o relație de forma:

$$P_{ij} = \begin{cases} \frac{(\tau_{ij})^\alpha \cdot (1/d_{ij})^\beta}{\sum_{p \in Tabu_k} (\tau_{ip})^\alpha \cdot (1/d_{ip})^\beta} & j \notin Tabu_k \\ 0 & j \in Tabu_k \end{cases} \quad (1)$$

Dintre toate nodurile  $j$  în care se admite deplasarea din nodul  $i$ , se alege nodul  $j^*$  pentru care probabilitatea  $P_{ij^*}$  este maximă. În consecință, agentul  $k$  se va deplasa în nodul  $j^*$  ( $Oras_k = j^*$ ), iar nodul  $j^*$  va fi introdus în lista tabu ( $Tabu_k(s) = j^*$ ). În relația (1)  $\alpha$  și  $\beta$  sunt doi parametri care controlează

ponderea cu care intervin concentrația de feromoni ( $\tau_{ij}$ ) și vizibilitatea ( $1 / d_{ij}$ ) în calculul probabilității  $P_{ij}$ . De exemplu, pentru  $\beta = 0$ , probabilitățile  $P_{ij}$  se stabilesc numai în funcție de concentrația de feromoni, iar pentru  $\alpha = 0$  – numai în funcție de vizibilitatea nodurilor (distanțele dintre noduri).

După parcurgerea a  $N$  pași, când toți agenții au trecut prin toate nodurile grafului, traseul fiecărui agent se încheie prin revenirea în nodul de origine. În continuare, se calculează lungimile traseelor pentru toți agenții și, dintre acestea, se memorează lungimea minimă. În acest moment se încheie un pas / o iterație a algoritmului.

Înainte de trecerea la pasul următor, se procedează la actualizarea concentrației de feromoni de pe fiecare latură a grafului. Aceasta se face o singură dată, la încheierea fiecărui ciclu / iterație, atunci când fiecare agent a parcurs un tur complet în graf. În acest caz, cantitatea de feromoni adăugată pe fiecare latură este invers proporțională cu lungimea întregului traseu parcurs de agentul respectiv. Ca urmare, acesta este modelul de actualizare a concentrațiilor de feromoni pe laturile grafului folosit în algoritmul prezentat în Caseta 1 și folosește o relație de forma:

$$\tau_{ij} = \rho \cdot \tau_{ij} + \Delta \tau_{ij} \quad (2)$$

unde  $\rho$  este un coeficient subunitar, astfel încât  $(1 - \rho)$  reprezintă rata de evaporare a feromonilor de pe traseele formate. Coeficientul  $\rho$  se alege întotdeauna subunitar, astfel încât să se evite acumularea nelimitată de feromoni pe laturile grafului (de exemplu,  $\rho = 0.1$ ).  $\Delta \tau_{ij}$  reprezintă corecția concentrației de feromoni pe latura  $(i, j)$ , determinată de numărul total de agenți care se deplasează din nodul  $i$  în nodul  $j$ . Aceste corecții se calculează cu relația:

$$\Delta \tau_{ij} = \sum_{k=1}^{NA} \Delta \tau_{ij}^k \quad (3)$$

unde  $\Delta \tau_{ij}^k$  este cantitatea de feromoni lăsată pe latura  $(i, j)$  de agentul  $k$ :

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{daca } i, j \in \text{Tabu}_k \text{ si } i = \text{Tabu}_k(p); j = \text{Tabu}_k(p+1) \\ 0 & \text{in caz contrar} \end{cases} \quad (4)$$

Criteriul de oprire folosit în cadrul algoritmului este atingerea unui număr maxim de iterații,  $T_{max}$ . Atât timp cât  $t < T_{max}$ , se reinițializează listele tabu ale agenților și se reia procedura prin reinițializarea primului element din fiecare lista tabu cu numărul de ordine al orașului în care se află fiecare agent.

## APLICAȚIE

În cadrul lucrării, cu ajutorul funcției ANTS, se va rula în MATLAB problema comisului voiajor pentru diverse valori ale parametrilor de intrare ale algoritmului furnicii.

Funcția solicită următorii parametri de intrare:

- $N$  – numărul de orașe ce trebuie parcurs de comisul voiajor.
- $NA$  – numărul de agenți (furnici) – un multiplu al numărului de orașe  $N$  ;
- $c$  – concentrația inițială de feromoni de pe trasee;
- $\rho$  – rata de evaporare a feromonilor;
- $\alpha, \beta$  – parametri de control ai ponderii concentrației de feromoni și ai vizibilității;
- $tmax$  – numărul maxim de iterații ale algoritmului
- $a$  – vector de forma  $[x_1, y_1, x_2, y_2, \dots, x_N, y_N]$  care codifică coordonatele spațiale ale orașelor vizitate de comisul voiajor.

Rezultatele algoritmului sunt următoarele:

- $tabopt$  - bucla cu distanta minima parcursa de un agent
- $lmin$  - lungimea totala a drumului optim, calculată conform listei tabu optime  $tabopt$  și a coordonatelor orașelor

Se vor trage concluzii referitor la performanțele algoritmului pentru diverse valori ale parametrilor  $N, NA, \alpha$  și  $\beta$ .